1. **Write a program for Abstraction?**

```java
abstract class Test {
    abstract String displayMessage(String name);

    abstract void displayMessage();
}

public class Abstraction extends Test {
    public static void main(String[] args) {
        Abstraction abstraction = new Abstraction();

        abstraction.displayMessage();
        System.out.println(abstraction.displayMessage("Test")); // Test

    }

    String displayMessage(String name) {
        return name;
    }

    void displayMessage() {
        System.out.println("Implementation for void method"); // Implementation for void method
    }
}
```

2. **Write an example for Encapsulation?**

```java
class Student {

    private int id;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

}

public class Encapsulation {

    public static void main(String[] args) {
        Student student = new Student();
        student.setId(10);

        System.out.println(student.getId()); //10
    }

}
```

3. **Write an example for Polymorphism?**

```
class C {

    public void display() {
        System.out.println("in Parent class Display");
    }
}

class D extends C {
    public void display() {
        System.out.println("in Parent class Display");
    }
}
public class Polymorphism {

    int sum(int i, int j) {
        return i + j;
    }

    int sum(int i, int j,int k) {
        return i + j + k;
    }

    public static void main(String[] args) {

        //Compile time Polymorphism
        Polymorphism polymorphism = new Polymorphism();
        System.out.println(polymorphism.sum(5, 10)); //15
        System.out.println(polymorphism.sum(4, 8,9)); //21

        //Runtime Polymorphisim
        C a = new C();
        a.display();
        D b = new D();
        b.display();
    }
}
```

4. **How to reverse a string without recursion?**

```
public static void main(String[] args) {
    String str = "Some Thing";

    String reverse = "";

    for (int i = str.length(); i > 0; --i) {
        reverse = reverse + (str.charAt(i-1));
    }

    System.out.println(reverse);//gnihT emoS

}
```

5. **How to swap two Strings without using a third variable?**

```
public static void main(String[] args) {
    String a = "abc";
    String b = "def";

    a = a + b;
    b = a.substring(0, a.length() - b.length());
    a = a.substring(b.length());

    System.out.println(a + " " + b); //def abc
}
```

6. **Write a program to reverse a number?**

```java
public static void main(String[] args) {
    int number = 126754;
    int reverse = 0;

    while (number != 0) {
        reverse = (reverse * 10) + (number % 10);
        number = number / 10;
    }

    System.out.println(reverse);//457621
}
```

7. **Write an example for Sting Sub String?**

```java
public static void main(String[] args) {

    String name = "TestingFramework";

    System.out.println(name.substring(3, 8)); //tingF

    name = "Testing Frame Work";

    System.out.println(name.substring(0, 9)); //Testing F
    System.out.println(name.substring(1, 9)); //esting F
}
```

8. **Write a program to check given input is palindrome or not?**

```java
public class PalindromExample {

    public static void main(String[] args) {

        String data = "aba";

        //Solution 1
        System.out.println(new StringBuilder().append(data).reverse().toString().equals(data));

        // Second Approach
        boolean isPalindrom = checkPalindrom(data);
        System.out.println(isPalindrom); // true
    }

    private static boolean checkPalindrom(String input) {
        String reverse = "";
        for (int i = input.length(); i > 0; --i) {
            reverse = reverse + (input.charAt(i - 1));
        }
        return reverse.equals(input);
    }

}
```

9. **Write an example to know Even or Odd number?**

```
public static void main(String[] args) {
    int num =10;

    if(num%2 == 0) {
        System.out.println("Even"); //Even
    } else{
        System.out.println("Odd");
    }

    num =11;

    if(num%2 == 0) {
        System.out.println("Even");
    } else{
        System.out.println("Odd");//Odd
    }
}
```

10. Write an example to know given number is prime or not?

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. For example, 5 is prime, as only 1 and 5 divide it, whereas 6 is composite, since it has the divisors 2 and 3 in addition to 1 and 6.

```
public static void main(String[] args) {

    System.out.println(isPrimeNumber(10)); //false
    System.out.println(isPrimeNumber(11)); //true
}

public static boolean isPrimeNumber(int number) {

    for (int i = 2; i <= number / 2; i++) {
        if (number % i == 0) {
            return false;
        }
    }
    return true;
}
```

11. Given an example to get the factorial of a number?

Example: if input is 3: 3*2*1 =6

```
public static void main(String[] args) {

    int fact = 1;
    int input = 3;

    for (int i = 1; i <= input; i++) {
        fact = fact * i;
    }

    System.out.println(fact); // 6
}
```

12. Write a logic to find the duplicate characters in String?

```
public static void main(String[] args) {

    String name = "SunnySy";

    Map<Character, Integer> dupCountMap = new HashMap<>();

    char[] nameArr = name.toCharArray();

    for (char c : nameArr) {
        if (dupCountMap.containsKey(c)) {
            dupCountMap.put(c, dupCountMap.get(c) + 1);
        } else {
            dupCountMap.put(c, 1);
        }
    }

    dupCountMap.forEach((k,v) -> System.out.println(k + " Count " + v)); //S Count 2 u Count 1  y Count 2 n Count 2
}
```

13. Write a program to check String is Anagram or not?

Two strings are called anagrams if they contain same set of characters but in different order.

```
public class AnagramExample {

    public static void main(String[] args) {
        boolean test = isAnagram("Keep", "Peek");
        System.out.println(test);
    }

    private static boolean isAnagram(String string, String string2) {
        char [] arr1 = string.toLowerCase().toCharArray();
        char [] arr2 = string2.toLowerCase().toCharArray();

        Arrays.sort(arr1);
        Arrays.sort(arr2);

        return Arrays.equals(arr1, arr2);
    }
```

14. Write a program to found the highest repeating character in String?

```
public class MaxRepeatingCharacters {

    public static void main(String[] args) {
        String str = "HeGGo";

        System.out.println(maxRepeating(str)); //G
    }

    private static char maxRepeating(String str) {
        int len = str.length();
        int count = 0;

        char res = str.charAt(0);

        for (int i = 0; i < len; i++) {
            int cur_count = 1;
            for (int j = i + 1; j < len; j++) {
                if (str.charAt(i) != str.charAt(j))
                    break;
                cur_count++;
            }

            // Update result if required
            if (cur_count > count) {
                count = cur_count;
                res = str.charAt(i);
            }
        }
        return res;
    }
```

15. Write a program to print fibonacci series?

By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

```
public static void main(String[] args) {
    int num1 = 0;
    int num2 = 1;
    int counter = 0;
    int input = 5;

    while (counter < input) {

        System.out.print(num1 + " "); //0 1 1 2 3

        int num3 = num2 + num1;
        num1 = num2;
        num2 = num3;
        counter = counter + 1;
    }
}
```

16. Write a program to print 1 to 100 and 100 to 1 numbers without loop?

```
// Sample Java Program to Print 1 to 100 without Loop
public class print1to100 {
        public static void main(String[] args)
        {
                int number = 1;

                printNumbers(number);
        }

        public static void printNumbers(int num)
        {
                if(num <= 100)
                {
                        System.out.print(num +" ");
                        printNumbers(num + 1);
                }
        }
}
```

```
// Sample Java Program to Print 100 to 1 without Loop
public class print1to100Ex2 {
        public static void main(String[] args)
        {
                printNumbers(100);
        }

        public static void printNumbers(int num)
        {
                if(num > 0)
                {
                        System.out.print(num +" ");
                        printNumbers(num - 1);
                }
        }
}
```

17. Write a logic to find the duplicate numbers and its count in an Array?

```
public static void main(String[] args) {

    var num= new int[] { 1, 2, 4, 9, 1, 2, 4, 3 };

    Map<Integer, Integer> dupMap = new HashMap<>();

    for (int i : num) {

        if (dupMap.containsKey(i)) {
            dupMap.put(i, dupMap.get(i) + 1);
        } else {
            dupMap.put(i, 1);
        }
    }

    dupMap.forEach((k,v) -> System.out.println(k +" Count "+ v)); //1 Count 2 2 Count 2 3 Count 1   4 Count 2 9 Count 1

}
```

18. Write a program to shuffle the array?

```
public class ShuffleArray {

    public static void main(String[] args) {
        Integer[] intArray = { 1, 2, 3, 4, 5, 6, 7 };

        List<Integer> intList = Arrays.asList(intArray);

        //Approach 1
        Collections.shuffle(intList);
        intList.stream().forEach(System.out::print); //5 4 3 7 1 6 2

        //Approach 2
        Random rand = new Random();
        for(int i=0;i<intArray.length;i++) {
            int randomIndexToSwap = rand.nextInt(intArray.length);
            int temp = intArray[randomIndexToSwap];
            intArray[randomIndexToSwap] = intArray[i];
            intArray[i] = temp;
        }

        Arrays.stream(intArray).forEach(System.out::print);//5 2 1 6 3 7 4

    }
```

19. Write a java program to rearrange Positive & Negative Values in an Array?

```java
public class NegativeArrays {

    public static void main(String[] args) {
        int[] arr = { 2, 8, 5, -5, -8, 9 };

        for (int i = 0; i < arr.length; i++) {
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[i] > arr[j]) {
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
        Arrays.stream(arr).forEach(System.out::println);//-8 -5 2 5 8 5
    }

}
```

20. Prove String is immutable?

```java
public class StringImmutable {

    public static void main(String[] args) {
        String s1 = "JAVA";

        String s2 = "JAVA";

        System.out.println(s1 == s2); // Output : true

        s1 = s1 + "J2EE";

        System.out.println(s1 == s2); // Output : false
    }
```

That means now both s1 and s2 are pointing to two different objects in the pool. Before modifications they are pointing to same object. Once we tried to change the content of the object using 's1', a new object is created in the pool with "JAVAJ2EE" as it's content and it's reference is assigned to s1. If the strings are mutable, both s1 and s2 should point to same object even after modification. That never happened here. That proves the string objects are immutable in java.

21. Write a program to find Highest and Lowest numbers in an array?

```java
public static void main(String[] args) {
    int numArr[] = new int[] { 1, 99, 34, 24, 65 };

    // Solution 1
    Arrays.sort(numArr);
    System.out.println("Highest Num " + numArr[numArr.length - 1] + " Lowest Num " + numArr[0]); // Highest Num 99

}
```

22. Give best example to compare and find common elements two arrays?

Below is the best way if we can manually compare it by writing the separate loops

```
public static void main(String[] args) {

    int numArr[] = { 1, 99, 34, 24, 65 };
    int numArr2[] = { 1, 87, 34, 64, 65 };

    // Checking both arrays are Equal
    Object[] arr1 = { numArr };
    Object[] arr2 = { numArr2 };

    if (Arrays.deepEquals(arr1, arr2)) {
        System.out.println("Equal");
    } else {
        System.out.println("Not Equal"); // Not Equal
    }

    // Finding out two common elements in two arrays
    Set<Integer> setOne = new HashSet<>();
    Arrays.stream(numArr).forEach(item -> setOne.add(item));

    Set<Integer> setTwo = new HashSet<>();
    Arrays.stream(numArr2).forEach(item -> setTwo.add(item));

    setOne.retainAll(setTwo);

    System.out.println(setOne); //[1, 65, 34]
```

23. Write a program to remove duplicates from sorted array?

```
public static void main(String[] args) {
    int[] arr = { 1, 2, 2, 3, 5, 4, 5, 4, 1 };

    Arrays.stream(arr).sorted().distinct().forEach(System.out::println); //1 2 3 4 5
}
```

24. Write a program to convert string to number without using Integer.parseInt() method?

```
public static void main(String[] args) {
    String input = "123456";
    char[] chArr = input.toCharArray();
    int sum = 0;

    // get ascii value for zero
    int zeroAscii = (int) '0';
    System.out.println(zeroAscii); //48
    for (char c : chArr) {
        int tmpAscii = (int) c;
        sum = (sum * 10) + (tmpAscii - zeroAscii);
    }

    System.out.println(sum); // 123456
}
```

25. Write a program to create deadlock between two threads?

```
public static void main(String[] args) {
    String str1 = "Java";
    String str2 = "UNIX";

    Thread trd1 = new Thread("My Thread 1") {
        public void run() {
            while (true) {
                synchronized (str1) {
                    synchronized (str2) {
                        System.out.println(str1 + str2);
                    }
                }
            }
        }
    };

    Thread trd2 = new Thread("My Thread 2") {
        public void run() {
            while (true) {
                synchronized (str2) {
                    synchronized (str1) {
                        System.out.println(str2 + str1);
                    }
                }
            }
        }
    };

    trd1.start();
    trd2.start();
```

26. How to swap two numbers without using temporary variable?

```
public class MySwapingTwoNumbers {

    public static void main(String[] args) {
        int x = 10;
        int y = 20;

        x = x + y;
        y = x - y;
        x = x - y;

        System.out.println(x + " " + y); //20 10
    }

}
```

27. Write a program to implement hashcode and equals.

```java
class Price {
    public String name;
    public int id;

    Price(String name, int id) {
        this.name = name;
        this.id = id;
    }

    @Override
    public boolean equals(Object obj) {
        Price price = (Price) obj;
        return (price.name == this.name && price.id == this.id);

    }

    @Override
    public int hashCode() {
        return this.id;
    }
}

public class HashCodeAndEquals {

    public static void main(String[] args) {
        Price g1 = new Price("aa", 1);
        Price g2 = new Price("aa", 1);

        if (g1.hashCode() == g2.hashCode()) {
            if (g1.equals(g2))
                System.out.println("Both Objects are equal. "); //Both Objects are equal.
            else
                System.out.println("Both Objects are not equal. ");
        } else
            System.out.println("Both Objects are not equal. ");
    }

}
```

28. Give an example for Bubble Sort?

```java
public class BubbleSort {

    public static void main(String[] args) {
        int arr[] = { 3, 60, 35, 2, 45, 320, 5 };

        int n = arr.length;
        int temp = 0;

        for (int i = 0; i < n; i++) {
            for (int j = 1; j < n - i; j++) {
                if (arr[j - 1] > arr[j]) {
                    temp = arr[j - 1];
                    arr[j - 1] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        Arrays.stream(arr).forEach(System.out::println); //2 3  5 35 45 60  320
```

29. Give an example for Binary Search?

```java
public class BinarySearch {
    public static void main(String[] args) {
        int arr[] = { 2, 3, 4, 10, 40 };
        int searchElement = 10;
        int left = 0;
        int right = arr.length - 1;

        int result = binarySearch(arr, left, right, searchElement);

        if (result == -1)
            System.out.println("Element not present");
        else
            System.out.println("Element found at index " + result);
    }

    static int binarySearch(int arr[], int left, int right, int x) {
        if (right >= left) {
            int mid = left + (right - left) / 2;
            System.out.println("mid " +mid);
            // If the element is present at the
            // middle itself
            if (arr[mid] == x)
                return mid;

            // If element is smaller than mid, then
            // it can only be present in left subarray
            if (arr[mid] > x)
                return binarySearch(arr, left, mid - 1, x);

            // Else the element can only be present
            // in right subarray
            return binarySearch(arr, mid + 1, right, x);
        }

        // We reach here when element is not present
        // in array
        return -1;
    }
```

30. Write a logic remove duplicates from ArrayList without using Set or Map?

```java
public class RemoveDuplicateList {

    public static void main(String[] args) {
        List<Integer> intList = new ArrayList<>();
        intList.add(1);
        intList.add(3);
        intList.add(3);
        intList.add(2);
        intList.add(1);

        intList.stream().distinct().forEach(System.out::println); //1 3 2
    }
```

31. Write a logic to remove the duplicate objects from List?

```java
public class RemoveDuplicateObjects {

    public static void main(String[] args) {
        Student student = new Student();
        student.setId(101);
        student.setName("Sunny");

        Student student1 = new Student();
        student1.setId(102);
        student1.setName("Bunny");

        Student student2 = new Student();
        student2.setId(101);
        student2.setName("Sunny");

        List<Student> studentList = new ArrayList<>();
        studentList.add(student);
        studentList.add(student1);
        studentList.add(student2);

        //Removing the duplicate objects
        Set<Student> distinctSet = studentList.stream()
                .collect(Collectors.toCollection(() -> new TreeSet<>(Comparator.comparing(Student::getId))));

        distinctSet.forEach(item -> System.out.println(item.getId() + " " + item.getName()));
    }
```

32. Given the list of employees, count number of employees with salary > 5000?

```java
public static void main(String[] args) {
    Student student = new Student();
    student.setId(101);
    student.setName("Sunny");
    student.setSalary(29000);;

    Student student1 = new Student();
    student1.setId(102);
    student1.setName("Bunny");
    student1.setSalary(5000);

    List<Student> studentList = new ArrayList<>();
    studentList.add(student);
    studentList.add(student1);

    //Removing the duplicate objects
    long salaryCount = studentList.stream().filter(emp -> emp.getSalary()>5000).count();
    System.out.println(salaryCount); //1
```

33. Given the list of employees, get the name of the employee with max and minimum salary?

```
Student student = new Student();
student.setId(101);
student.setName("Sunny");
student.setSalary(29000);

Student student1 = new Student();
student1.setId(102);
student1.setName("Bunny");
student1.setSalary(5000);

Student student2 = new Student();
student2.setId(103);
student2.setName("Munny");
student2.setSalary(1000);

List<Student> studentList = new ArrayList<>();
studentList.add(student);
studentList.add(student1);
studentList.add(student2);

//Max Salary
Student maxSalaryEmp = studentList.stream().max(Comparator.comparing(Student::getSalary)).get();
System.out.println(maxSalaryEmp.getName()); //Sunny

//Minimum Salary
Student minSalaryEmp =studentList.stream().min(Comparator.comparing(Student::getSalary)).get();
System.out.println(minSalaryEmp.getName()); //Munny
```

34. Write a logic to sort employee by his Name?

```
Student student = new Student();
student.setId(101);
student.setName("Sunny");
student.setSalary(29000);

Student student1 = new Student();
student1.setId(102);
student1.setName("Bunny");
student1.setSalary(5000);

Student student2 = new Student();
student2.setId(103);
student2.setName("Munny");
student2.setSalary(1000);

List<Student> studentList = new ArrayList<>();
studentList.add(student);
studentList.add(student1);
studentList.add(student2);

//Sorting by name
List<Student> sortedList = studentList.stream().sorted(Comparator.comparing(Student::getName)).collect(Collectors.toList());

sortedList.forEach(item -> System.out.println(item.getId() + "  " + item.getName()));//102  Bunny   103  Munny  101  Sunny
```

35. Given the list of employee, group them by employee name?

```
public static void main(String[] args) {
    Student student = new Student();
    student.setId(101);
    student.setName("Sunny");
    student.setSalary(29000);

    Student student1 = new Student();
    student1.setId(102);
    student1.setName("Bunny");
    student1.setSalary(5000);

    Student student2 = new Student();
    student2.setId(103);
    student2.setName("Munny");
    student2.setSalary(1000);

    List<Student> studentList = new ArrayList<>();
    studentList.add(student);
    studentList.add(student1);
    studentList.add(student2);

    //Sorting by name
    Map<String, List<Student>> sortedList = studentList.stream().collect(Collectors.groupingBy(Student::getName));

    sortedList.forEach((k,v) ->{
        v.forEach(data -> System.out.println("Name "+ k + " Empi Id " + data.getId() + " Emp Name " + data.getName()));
        /*
         Name Munny Empi Id 103 Emp Name Munny
         Name Bunny Empi Id 102 Emp Name Bunny
         Name Sunny Empi Id 101 Emp Name Sunny

         */
    });
```

36. Write a program to display the employee details in list descending order?

```
public static void main(String[] args) {
    List<Integer> intList = new ArrayList<>();
    intList.add(1);
    intList.add(9);
    intList.add(2);

    Collections.reverse(intList);

    System.out.println(intList);

    List<Integer> nwList = intList.stream().sorted().collect(Collectors.toList());

    System.out.println(nwList);

    List<Employee> empList = new ArrayList<>();
    Employee employee = new Employee();
    employee.setName("Bunny");
    employee.setSalary(1000);
    employee.setEmpId(100);
    empList.add(employee);

    Employee employee1 = new Employee();
    employee1.setName("Sunny");
    employee1.setSalary(2000);
    employee1.setEmpId(102);
    empList.add(employee1);

    Employee employee2 = new Employee();
    employee2.setName("Munny");
    employee2.setSalary(5000);
    employee2.setEmpId(101);
    empList.add(employee2);

    //Sorting Employee in descending order
    empList.sort(Comparator.comparing(Employee::getEmpId).reversed());

    empList.forEach(empData -> System.out.println(empData.getEmpId()));
```

37. Write a program to display the employee details in map descending order?

```
public static void main(String[] args) {
    Map<Integer, Employee> unsortMap = new HashMap<>();

    Employee employee = new Employee();
    employee.setName("Sunny");
    employee.setSalary(2000);
    employee.setEmpId(102);
    unsortMap.put(employee.getEmpId(), employee);

    Employee employee1 = new Employee();
    employee1.setName("Bunny");
    employee1.setSalary(1000);
    employee1.setEmpId(100);
    unsortMap.put(employee1.getEmpId(), employee1);

    Employee employee2 = new Employee();
    employee2.setName("Munny");
    employee2.setSalary(5000);
    employee2.setEmpId(101);
    unsortMap.put(employee2.getEmpId(), employee2);

    LinkedHashMap<Integer, Employee> sortedMapByKey = new LinkedHashMap<>();

    //Sorting Employee in descending order
    unsortMap.entrySet().stream().sorted(Map.Entry.comparingByValue(Comparator.comparing(Employee::getEmpId).reversed()))
            .forEachOrdered(data -> sortedMapByKey.put(data.getKey(), data.getValue()));

    sortedMapByKey.forEach((k,v)->System.out.println(v.getEmpId())); //102 101 100
```

38. Write a program to implement your own HashMap?

```
@Getter
@Setter
class MyEntry<K, V> {
    K key;
    V value;
    MyEntry<K, V> next;

    public MyEntry(K key, V value, MyEntry<K, V> next) {
        this.key = key;
        this.value = value;
        this.next = next;
    }

    @Override
    public int hashCode() {
        return (int) key;
    }

    @Override
    public boolean equals(Object obj) {
        return key.equals(obj);
    }
}

class HashMapImp<K, V> {
    List<MyEntry<K, V>> entryList = new ArrayList<>();

    public void put(K key, V value) {
        MyEntry<K, V> entry = new MyEntry<>(key, value, null);

        if (entryList.contains(entry.getKey())) {
            entryList.remove(entry.getKey());
        } else {
            entryList.add(entry);
        }
    }
```

```
public MyEntry<K, V> get(K key) {
    for (MyEntry<K, V> data : entryList) {
        if (data.getKey() == key) {
            return data;
        }
    }
    return null;
}
```

39. Write a logic to implement custom ArrayList?

```
public class CustomArrayList {
    private Object[] store;
    private int currentSize = 0;

    public CustomArrayList() {
        store = new Object[10];
    }

    public void add(Object input) {
        if (store.length - currentSize < 5) {
            store = Arrays.copyOf(store, store.length * 2);
        } else {
            store[currentSize++] = input;
        }
    }

    public Object get(int index) {
        Object data = null;
        if (index < currentSize) {
            data = store[index];
        } else {
            throw new ArrayIndexOutOfBoundsException();
        }
        return data;
    }

    public Object remove(int index) {
        if (index < currentSize) {
            store[index] = null;
        } else {
            throw new ArrayIndexOutOfBoundsException();
        }
        return store;
    }

    public int size() {
        return currentSize;
    }

public static void main(String[] args) {
    CustomArrayList mal = new CustomArrayList();
    mal.add(2);
    mal.add(5);
    for(int i=0;i<mal.size();i++){
        System.out.print(mal.get(i)+" "); //2 5
    }

    System.out.println("Element at Index 1:"+mal.get(1)); //Element at Index 1:5
    System.out.println("Removing element at index 1: "+mal.remove(1));

    for(int i=0;i<mal.size();i++){
        System.out.print(mal.get(i)+" "); //2 null
    }

}
```

40. Write a logic for inter thread communication?

```
public class WaitNotifyTest {
    private static final long SLEEP_INTERVAL = 3000;
    private boolean running = true;
    private Thread thread;

    public void start() {
        print("Inside start() method");
        thread = new Thread(new Runnable() {
            @Override
            public void run() {
                print("Inside run() method");
                try {
                    Thread.sleep(SLEEP_INTERVAL);
                } catch (InterruptedException e) {
                    Thread.currentThread().interrupt();
                }
                synchronized (WaitNotifyTest.this) {
                    running = false;
                    WaitNotifyTest.this.notify();
                }
            }
        });
        thread.start();
    }

    public void join() throws InterruptedException {
        print("Inside join() method");
        synchronized (this) {
            while (running) {
                print("Waiting for the peer thread to finish.");
                wait(); // waiting, not running
            }
            print("Peer thread finished.");
        }
    }
```

```
private void print(String s) {
    System.out.println(s);
}

public static void main(String[] args) throws InterruptedException {
    WaitNotifyTest test = new WaitNotifyTest();
    test.start();
    test.join();
    //Output
    /*
    Inside start() method
    Inside join() method
    Waiting for the peer thread to finish.
    Inside run() method
    Peer thread finished.

    */
}
```

41. Write a java 8 program to count the repeated words in String?

```
public class WordsCount {

    public static void main(String[] args) {

        String str = "aba,abc,aba,abc";
        List<String> list = Arrays.asList(str.split(","));

        //Case 1
        Map<String, Integer> countMap = list.stream().collect(Collectors.toMap(w -> w,w->1, Integer::sum));
        countMap.forEach((k,v)->System.out.println(k + " "+ v)); //aba 2 abc 2

        //Case 2
        Map<String, Long> countMap2 = list.stream().collect(Collectors.groupingBy(Function.identity(),Collectors.counting()));
        countMap2.forEach((k,v)->System.out.println(k + " "+ v));  //aba 2 abc 2
```

42. Given example for Singleton class that should not support multithreading access Deserialization, Clone, Reflection and overriding?

```java
//final class can not be extended
final class SigletonXyz {

    private static final SigletonXyz instance = null;

    private SigletonXyz() {

        // To prevent the Reflection
        if (instance != null) {
            throw new RuntimeException("Instance already available");
        }

    }

    // To stop object deserilization
    protected Object readResolve() {
        return instance;
    }

    // TO Stop the cloning
    @Override
    protected Object clone() throws CloneNotSupportedException {
        throw new CloneNotSupportedException();
    }

    // Making Thread safe
    public static synchronized SigletonXyz getInstance() {
        if (instance == null) {
            return new SigletonXyz();
        }
        return instance;
    }
}
```

43. How ConcurrentHashMap will work internally?

- ✓ The ConcurrentHashMap class is introduced in JDK 1.5 belongs to java.util.concurrent package.
- ✓ The underlined data structure for ConcurrentHashMap is Hashtable.
- ✓ ConcurrentHashMap class is thread-safe i.e. multiple threads can operate on a single object without any complications.
- ✓ At a time any number of threads are applicable for a read operation without locking the ConcurrentHashMap object which is not there in HashMap.
- ✓ In ConcurrentHashMap, the Object is divided into a number of segments according to the concurrency level.
- ✓ The default concurrency-level of ConcurrentHashMap is 16.
- ✓ In ConcurrentHashMap, at a time any number of threads can perform retrieval operation but for updated in the object, the thread must lock the particular segment in which the thread wants to operate. This type of locking mechanism is known as

Segment locking or bucket locking. Hence at a time, 16 update operations can be performed by threads.

- ✓ Inserting null objects is not possible in ConcurrentHashMap as a key or value.

44. Difference between HashTable, ConcurrentHashMap and SynchronizedMap?

ConcurrentHashMap	HashTable	SynchronizedMap
We will get thread safety without locking the total map object just with a bucket level lock.	We will get thread safety by locking the whole map object	We will get thread safety by locking the whole map object.
At a time multiple threads are allowed to operate on map objects safely.	At a time one thread is allowed to operate on a map object.	At a time only one thread is allowed to perform any operation on a map object.
Iterator of ConcurrentHashMap is fail-safe and won't raise ConcurrentModificationExcepti on	Iterator of HashTable is fail-fast and it will raise ConcurrentModificationExcepti on	Iterator of SynchronizedMap is fail-fast and it will raise ConcurrentModificationExcepti on
Null is not allowed for both keys and values.	Null is not allowed for both keys and values	Null is allowed for both keys and values.